

# Arquitectura de computadores

---

Pau Arlandis Martinez

## Sobre las normas de la asignatura

### Teoría

4 Prácticas (voluntarias). Solo puedes presentarte una vez.

- Entrada/Salida → Simulador 15%
- Memoria caché
- Pipeline de instrucciones } MC 88110 35%
- Multiprocesadores → C 25%

Las prácticas pueden sumar, juntas, hasta un punto más en la nota de teoría.

3 parciales

- Tema 1 → 30%
- Tema 2 → 40%
- Tema 3 y 4 → 40% (Posibilidad de recuperar tema 1 [25%] o tema 2 [35%])

Debe obtenerse una nota mínima de 2 puntos en cada uno para que hagan media.

### Práctica

1 proyecto (hasta la semana 9)

- Entrada/ Salida.
- Examen (al terminar, semana 9).
- Memoria escrita.

### Nota final

Siempre que ambas partes (teoría y proyecto) estén aprobadas (con más de un 5 cada una) la nota final se calcula mediante la fórmula:

$$0.7 \times nota_{teoría} + 0.3 \times nota_{proyecto}$$

## Tema 1 – Entrada / Salida

13 horas – Profesor: Manuel Nieto

### Introducción

#### Periféricos

Existen multitud de periféricos, con diferentes características cada uno:

- Modo de funcionamiento.
- Formato y tamaño de los datos. (Cuántos se intercambian, si este tamaño está definido...)
- Velocidad.
- Tiempo de acceso.

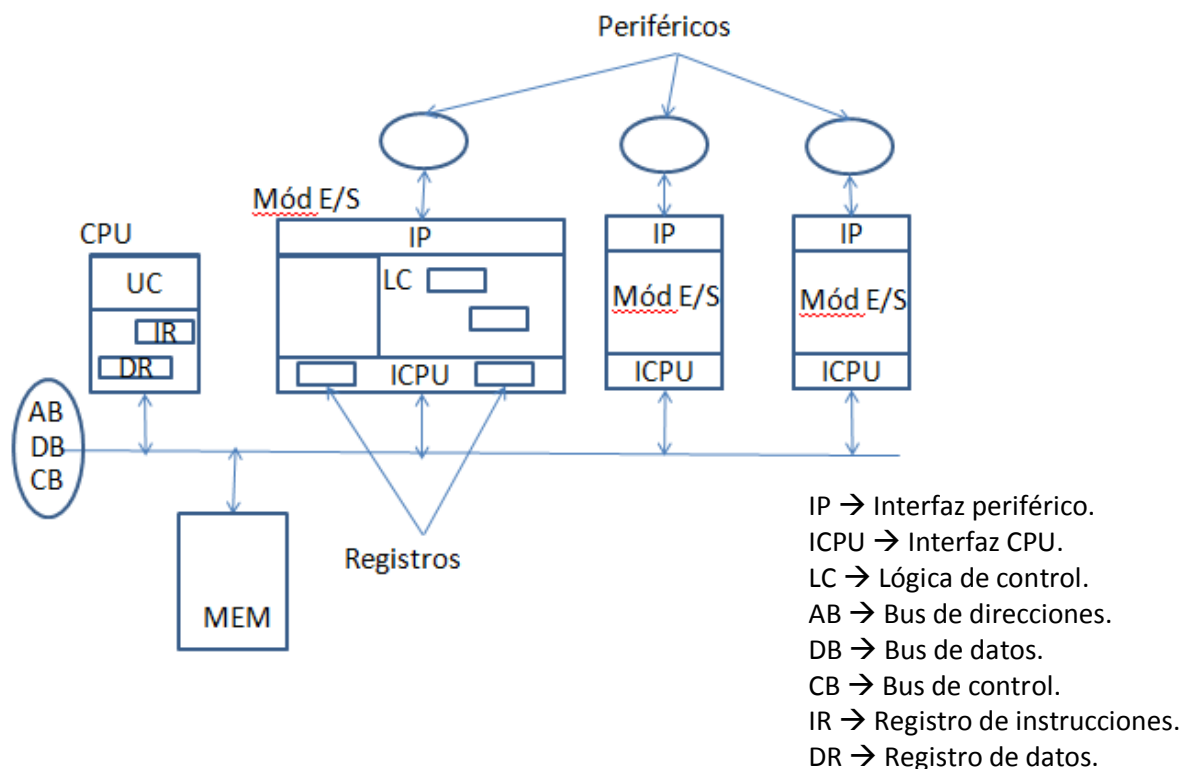
Se hace necesaria **unificar** la visión hardware de los periféricos.

Para ello nace el **módulo de E/S** que oculta las particularidades de cada periférico. La CPU solo dialoga con los módulos, que son todos iguales, estándar. Por tanto, tiene dos interfaces: Una que se comunica con el periférico y otra con el CPU.

Tiene varias funciones:

- Control...
- ...
- Comunicación con el periférico → siguiendo las características del mismo.
- Buffering. Dado que cada periférico tiene una velocidad diferente el módulo tiene capacidad de almacenamiento para poder controlar dicha velocidad y adaptarla a la de la CPU.
- Control de errores y situaciones anómalas.

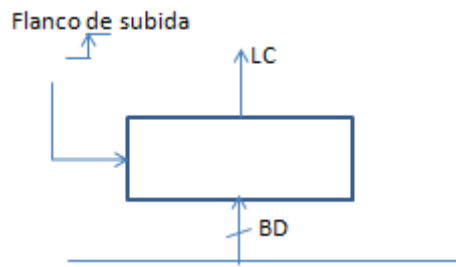
La estructura de un módulo de E/S sería algo similar a:



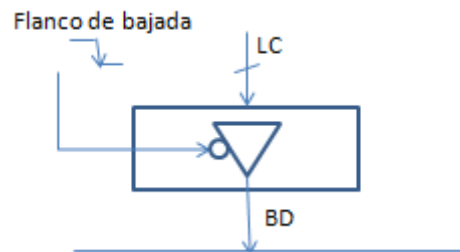
Lo único que controlamos son los registros de la Interfaz con la CPU. Estos son de tres tipos:

### Registros de datos

- **Registros de salida.** Un conjunto de biestables que nos permiten sacar información desde el bus de datos a la lógica de control.



- **Registro de entrada.** Saca información de la LC y la vuelca en el bus de datos. La salida se controla mediante un buffer triestado con estados 0, 1 y nada, vacío.



### Registros de estado

Son los que indican el estado del periférico y permiten el control de errores.

### Registros de control

Se encargan del control de todos los componentes y registros del módulo.

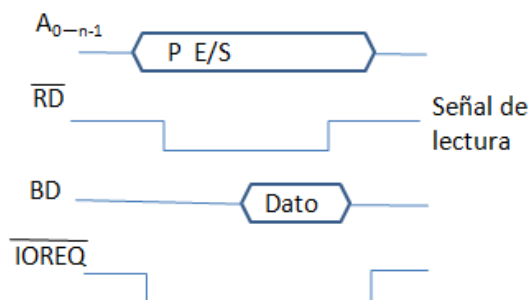
### Instrucciones de los módulos

Existen dos formas de dirigir el control de los módulos y de mapear sus direcciones.

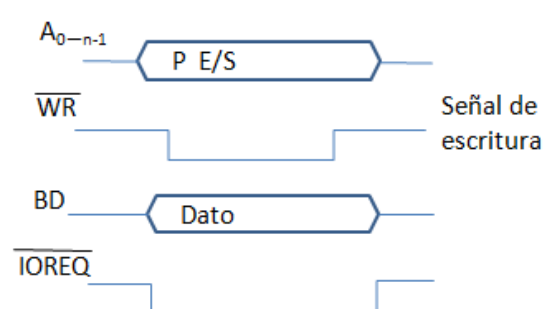
#### Mapas separados

Para controlar los módulos E/S con este método es necesario establecer un par de ciclos nuevas que permitan la entrada y la salida desde el procesador a los módulos.

#### Ciclo de entrada

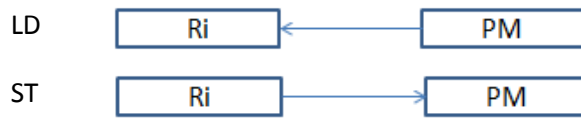


#### Ciclo de salida



Para que la memoria codifique, lea o modifique la información del bus se activa la señal  $\overline{\text{MEMRQ}}$ , para los módulos de E/S haremos igual. Crearemos la señal  $\overline{\text{IOREQ}}$ , si no está activa no hace nada la CPU, pero si lo está inicia la ejecución en el módulo.

A nivel de instrucciones sucede lo mismo, existen dos instrucciones para acceder a la memoria existen dos instrucciones:

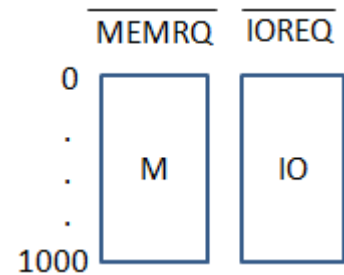


De forma paralela trabajaremos con los módulos, en los que existirán dos instrucciones:



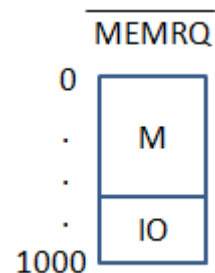
En este caso tendremos dos rangos de direcciones diferentes:

- 0-1000 (por ejemplo) con  $\overline{\text{MEMRQ}}$  activada.
- 0-1000 con  $\overline{\text{IOREQ}}$  activada.



### Mapeado en memoria

En este caso solo existe un rango de direcciones ya que no se dispone de señal  $\overline{\text{IOREQ}}$ , entonces no existen las instrucciones IN y OUT, solo las de direccionamiento de memoria. La solución que se toma es reservar un subrango de direcciones de la memoria para las de E/S. Debe tomarse un rango al principio o al final para mejorar la eficiencia.



Apuntes de Arquitectura de Computadores by [Pau Arlandis Martínez](#) is licensed under a [Creative Commons Reconocimiento 3.0 Unported License](#).